

# Versioning

---



**Marcel de Vries**

CTO

@marcelv | <http://fluentbytes.com>



# Outline



Versioning

Dependency hell

Semantic versioning

Continuous delivery and package management



# Versioning

---





Version numbers are an engineering artifact and should never be dictated from elsewhere!

-- Marcel de Vries



“

In the world of software management there exists a dread place called “dependency hell.”

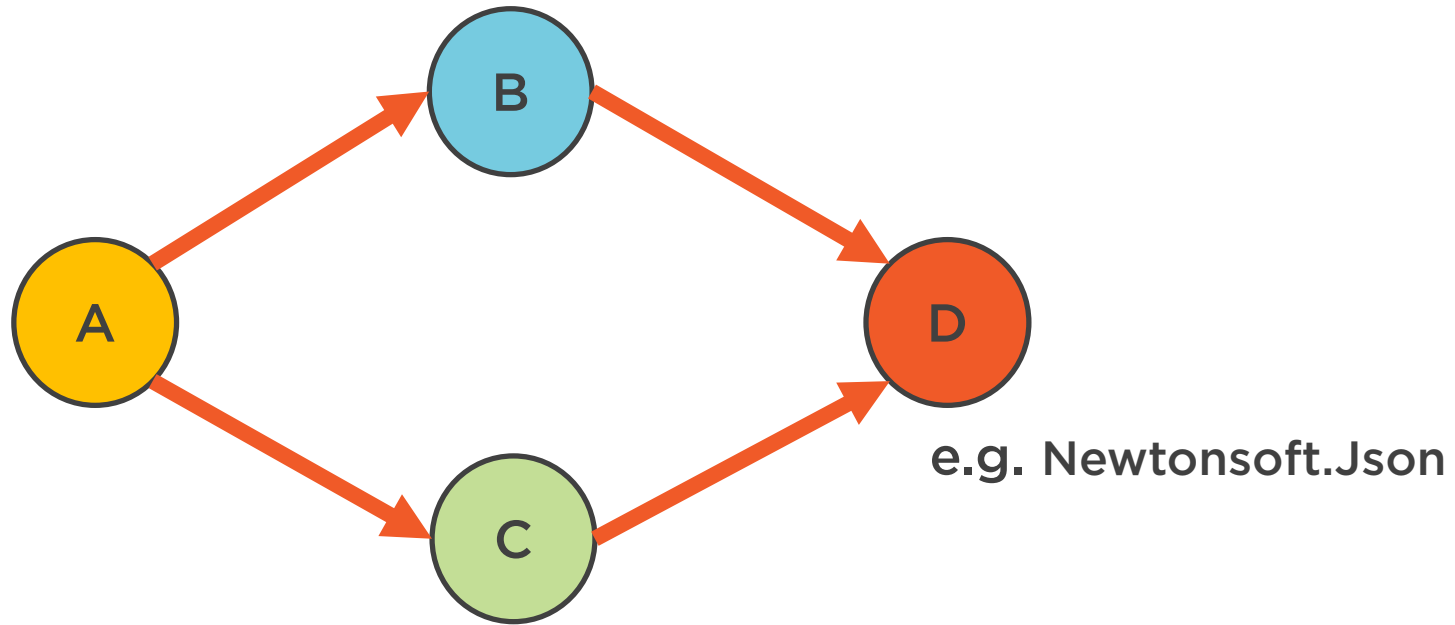
**Tom Preston-Werner**

”



# Dependency Management

## Dependency Hell 101



# Semantic Versioning

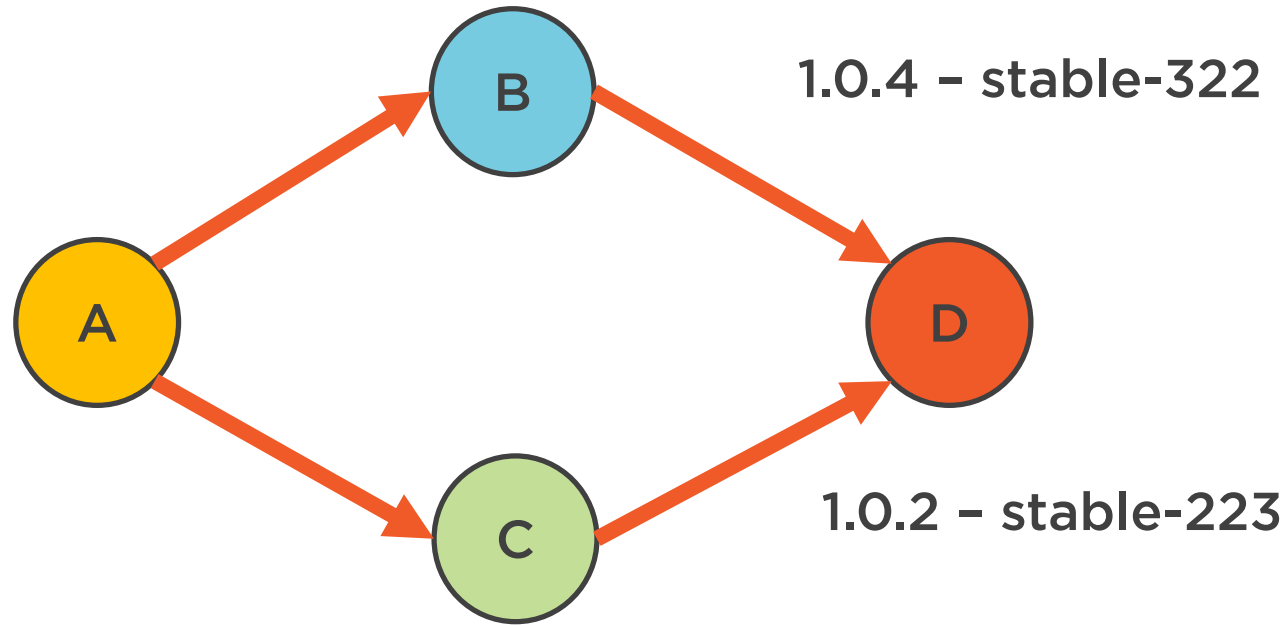


Given a version number **MAJOR.MINOR.PATCH**, increment the:

- **MAJOR** version when you make incompatible API changes,
- **MINOR** version when you add functionality in a backwards-compatible manner, and
- **PATCH** version when you make backwards-compatible bug fixes.
- Additional labels for pre-release and build metadata are available as extensions to the **MAJOR.MINOR.PATCH** format.

# Dependency Management

## Dependency Hell 101



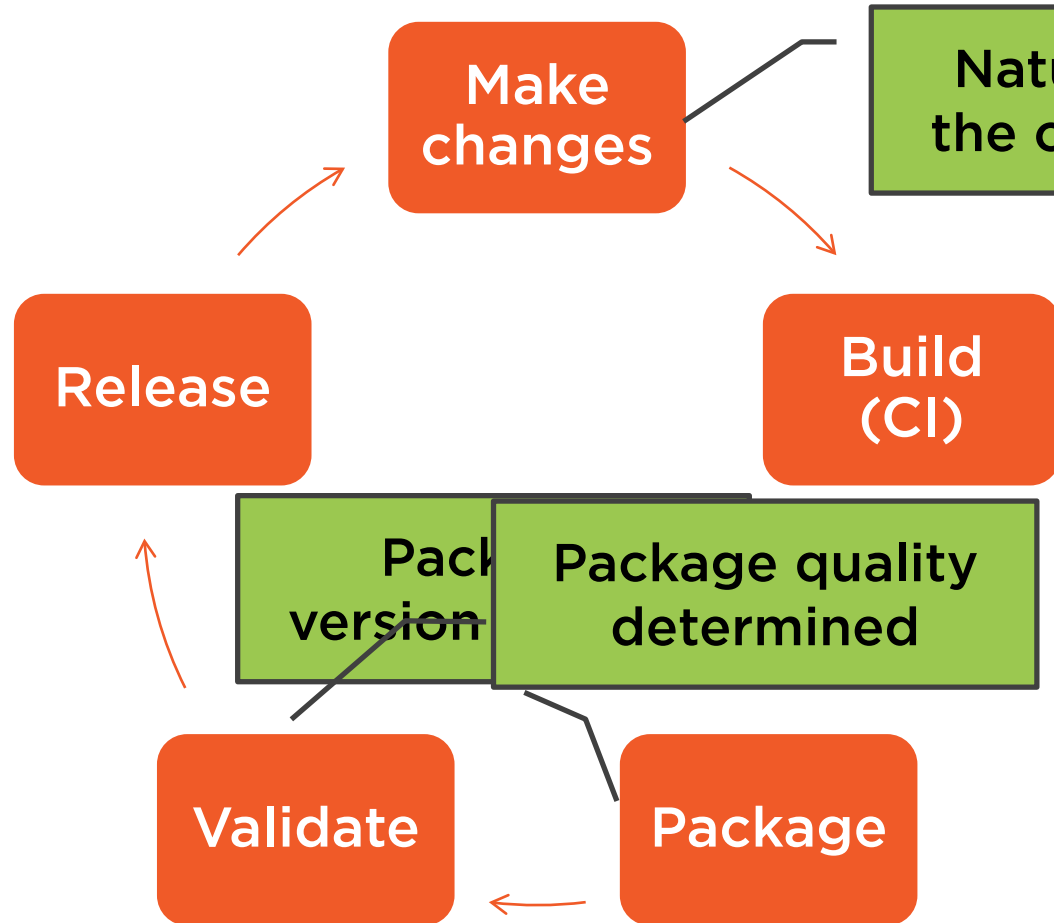


# Version Numbers and Packages

---



# Nature, Risk and Quality of a Change



Determine the nature, risk and quality

1.2.3 -beta1

Nature and compatibility of change

Quality of change

Communicate the nature, risk and quality



# Packages and Versioning



**maven**



Use semantic versioning for your packages

Combine semantic versioning and Gitflow branching strategy

- Determine version and communicated quality based on branch

Use a tool called GitVersion

Use the prerelease tag to communicate the quality and guarantee uniqueness of your package



# Demo



## Semantic Git versioning and NuGet



# Demo



## Semantic Git Versioning and NuGet With GitHub Actions



# Summary



Versioning

Dependency hell

Semantic versioning

Continuous delivery and package management

